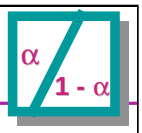
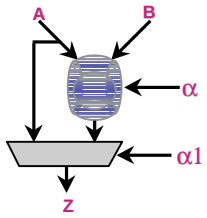
Linear Interpolation - Alpha Blender



- **Alpha Blender, used in graphics and video for pixel blending**



- **Definition: Z = Aα + B(1-α), range of α is [0,1]**
  A and B are "blended" together; α represents the "coverage" of A.
  - **If α = 0; Z = B**
  - **If α = 1; Z = A**
  - **If 0 < α < 1; Z = blend A & B with ratio defined by α**

1-1

•Common applications for alpha blending

  •mixing 2 or more video streams (2 at a time)

  •graphics rendering

•The picture in the upper right corner illustrates how alpha represents the "coverage" of the pixel

  •a number from 0 to 1

•The definition provides the equation. From this the degenerate cases where alpha equals 0 or 1 can be seen.

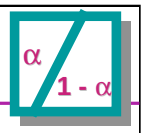
•When alpha is strictly between 0 and 1, the output is some "combination" of A and B. That's where "blending" comes from.

•In this particular design, 0 <= alpha < 1

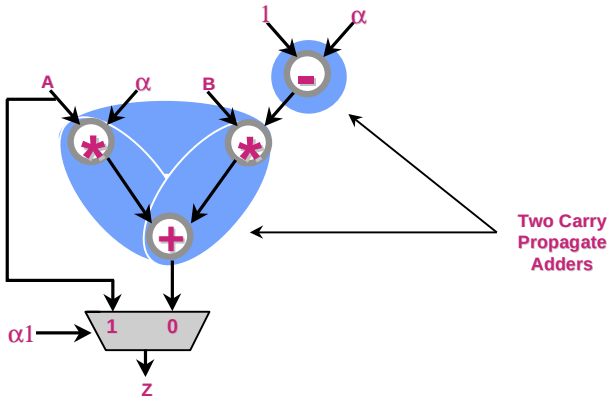
•alpha1 is used to indicate the value of 1.0

  •a bypass mux is used for this case

Alpha Blender: Architecture 0



- $Z = A\alpha + B(1-\alpha)$



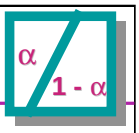
1-2

- This architecture is a direct mapping of the equation. It has
  - a subtractor for (1-alpha)
  - followed by a sum of two products

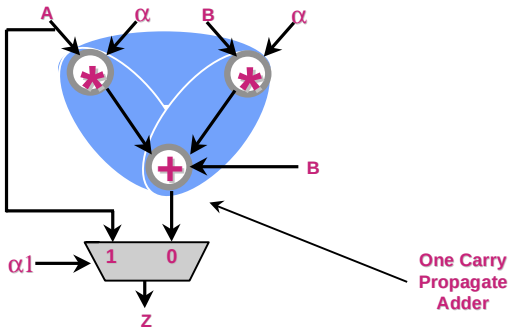
- There are two carry propagate adders in series
  - the subtractor
  - the carry propagate adder at the end of the merged sum of products

- There are two variable multipliers

Alpha Blender: Architecture 1



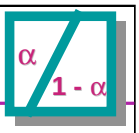
- $Z = A\alpha - B\alpha + B$



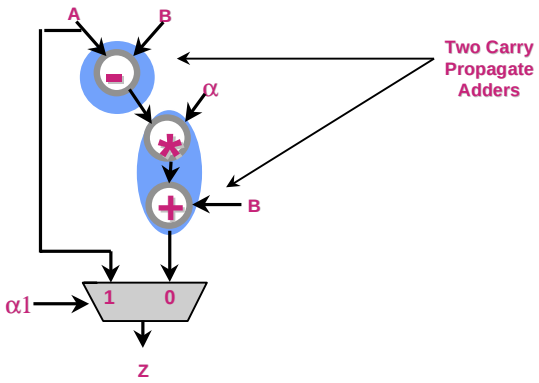
1-3

- This architecture comes from an algebraic manipulation of the equation
- It has two variable multipliers
- The sum of two products, plus the addition of B are merged together
  - there is only one carry propagate adder
- This architecture is smaller and faster than arch 0

Alpha Blender: Architecture 2



- $Z = (A - B)\alpha + B$



1-4

•This architecture comes from another algebraic manipulation

•It has a subtractor, followed by a merged multiply-add

- •there are two carry propagate adders in series

•Since there is only one multiplier, this architecture is smaller than arch 1

•Most people would expect this architecture to be slower than arch 1 because it has two carry propagations in series

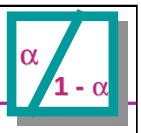
•The advanced datapath designer should note the following

- •there are half as many partial products (smaller carrysave tree)
- •booth encoding of alpha is in parallel with the subtractor

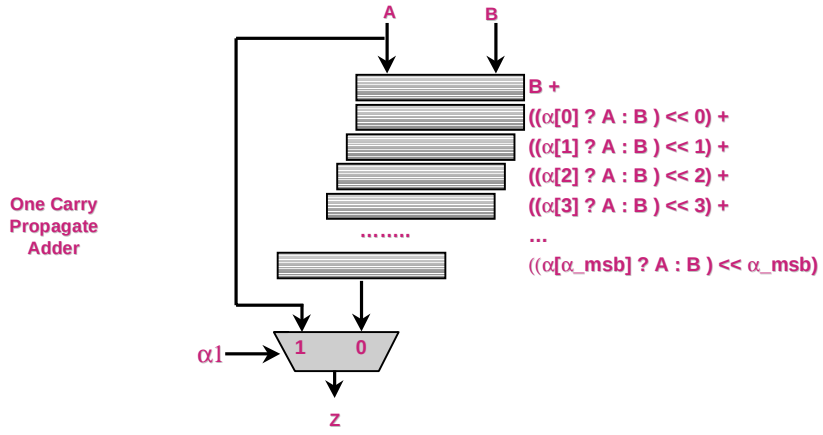
•For moderate bit widths this architecture is not only smaller than arch 1, it is also *faster* than arch 1

•For larger bit widths arch 1 becomes faster than this architecture

Alpha Blender: Architecture 3



- **Because α and (1 - α) are highly correlated, another architecture exists.**
- **Z = Aα + B*(~α) + B*2^-α_width**



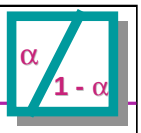
1-5

•This architecture comes from exploiting the correlation between alpha and (1-alpha). A derivation is provided in the MCL code.

•This architecture consists of a bunch of muxes driving a vector adder (carrysave tree followed by carry propagate adder).

•This architecture is smaller and faster than all the others

Alpha Blender: Architecture 4



- **gfxBlend() built in MC function**
- **See also**
  - gfxBit()
  - gfxLogicop()
  - gfxShift()

1-6

- MCL includes the superior architecture as a language primitive
  - it is a built-in function